# MMColor

MMColor_Configure
MMColor_CreateColorImage
MMColor_DecToHex
MMColor_GetColor
MMColor_GetColorComponent
MMColor_GetGrayscale
MMColor_GetImageColor
MMColor_HexToDec
MMColor_IsWebSafe
MMColor_Register
MMColor_SetColorComponent
MMColor_SetLayoutPartHeights
MMColor_ShowColorPicker
MMColor_ShowColorSpaceSlider
MMColor_ShowSlider
MMColor_Version
MMColor_VersionAutoUpdate
Reference_ColorSpace
Reference_Component
Reference_HexColor

# MMColor_Configure

### Description

Calling this function with no parameters will open the MMColor Configuration Dialog. You can optionally open to a specific tab by specifying the name of the tab as the "Option" parameter. This function also allows you to get or set any preference found in the Configuration Dialog.

Valid PrefNames:

"AddHelp" - If True, Function specific Help will be added to Calculations when inserting the Plug-in's Functions.

### Return Type

Varies

### Format

MMColor_Configure ( **Option** ; **PrefName** ; **PrefValue** )

### Optional Parameters

**Option**
Specify the name of a Tab in the Configuration Dialog to show it opened to that tab.
Specify "Get" with the PrefName parameter to get a preference value. (If not found, and PrefValue is defined, PrefValue will be returned.)
Specify "Set" with the PrefName and PrefValue parameters to set a preference value.

**PrefName**
The Name of the Preference to Get or Set. (See the Function Description for a list of valid PrefNames.)

**PrefValue**
The Value of the Preference to Set. (See the Function Description for some possible values.)

### Examples

# Example 1

## Code:

```
MMColor_Configure
```

## Result:

Opens the MMColor Configuration Dialog. (Because the "Option" parameter is not used, the dialog will open to the "Basics" tab.)

# Example 2

## Code:

```
MMColor_Configure( "About" )
```

## Result:

Opens the MMColor Configuration Dialog to the "About" tab.

# Example 3

## Code:

```
MMColor_Configure( "Get" ; "AddHelp" )
```

## Result:

Returns the value of the 'Add Help Comments to External Functions' setting from the Configuration Dialog.

# Example 4

## Code:

```
MMColor_Configure( "Set" ; "AddHelp" ; "False" )
```

## Result:

Sets the 'Add Help Comments to External Functions' setting in the Configuration Dialog to false.

# MMColor_CreateColorImage

### Description

This function takes a HexColor and creates a container image of the specified width and height (defaulting to 25x25) and returns that image to you. The native image type is a PNG (which allows for Alpha Transparency), but a JPEG "Preview" is also included in the container object for printing and using with Instant Web Publishing.

### Return Type

Container

### Format

MMColor_CreateColorImage ( **HexColor** ; **Width** ; **Height** ; **Alpha** )

### Required Parameters

**HexColor**
The Color of the Image to Create.

### Optional Parameters

**Width**
The Width of the Image. (Default is 25.)

**Height**
The Height of the Image. (Default is 25.)

**Alpha**
The Alpha/Transparency level. (0-255; Default is 255.)

### Related Items

MMColor_GetImageColor

### Examples

# Example 1

## Code:

```
MMColor_CreateColorImage( "FF00FF" )
```

## Result:

Returns a bright purple image.

# Example 2

## Code:

```
MMColor_CreateColorImage( "757500" ; 100 ; 50 ; 128 )
```

## Result:

Returns a 1/2 opaque olive image that is 100 pixels wide and 50 pixels tall.

# MMColor_DecToHex

### Description

Returns the Hex representation of the Decimal number entered in the "Dec" parameter, optionally with leading-zeros depending on the "NumDigits" parameter.

### Return Type

Text (HexColor)

### Format

MMColor_DecToHex ( **Dec** ; **NumDigits** )

### Required Parameters

**Dec**
The Decimal number to convert to Hexidecimal.

### Optional Parameters

**NumDigits**
The minimum number of digits in the returned Hexidecimal number.

### Related Items

MMColor_HexToDec

### Example

## Code:

```
MMColor_DecToHex( 128 )
```

## Result:

Returns 80.

# MMColor_GetColor

### Description

This function takes individual Component values and returns the HexColor equivalent for the specified Color Space.

### Return Type

Text (HexColor)

### Format

MMColor_GetColor ( **ColorSpace** ; **Component1** ; **Component2** ; **Component3** ; **Component4** )

### Required Parameters

**ColorSpace**
The Color Space that defines the remaining Component Parameters. Use "RGB", "HSV", "HSL", "CMY", or "CMYK".

**Component1**
Depending on the ColorSpace, specify the following Component Value:

"RGB" Color Space - Specify the Red Component. (0-255)

"HSV" Color Space - Specify the Hue Component. (0-359)

"HSL" Color Space - Specify the Hue Component. (0-359)

"CMY" Color Space - Specify the Cyan Component. (0-100)

"CMYK" Color Space - Specify the Cyan Component. (0-100)

**Component2**
Depending on the ColorSpace, specify the following Component Value:

"RGB" Color Space - Specify the Green Component. (0-255)

"HSV" Color Space - Specify the Saturation Component. (0-100)

"HSL" Color Space - Specify the Saturation Component. (0-100)

"CMY" Color Space - Specify the Magenta Component. (0-100)

"CMYK" Color Space - Specify the Magenta Component. (0-100)

**Component3**
Depending on the ColorSpace, specify the following Component Value:

"RGB" Color Space - Specify the Blue Component. (0-255)

"HSV" Color Space - Specify the Value Component. (0-100)

"HSL" Color Space - Specify the Luminance Component. (0-100)

"CMY" Color Space - Specify the Yellow Component. (0-100)

"CMYK" Color Space - Specify the Yellow Component. (0-100)

**Component4**
Depending on the ColorSpace, specify the following Component Value:

"CMYK" Color Space - Specify the Black Component. (0-100)

**Related Items**

MMColor_GetColorComponent, MMColor_SetColorComponent

**Examples**

# Example 1

## Code:

```
MMColor_GetColor( "RGB" ; 128 ; 0 ; 255 )
```

## Result:

Returns "8000FF"

# Example 2

## Code:

```
MMColor_GetColor( "HSV" ; 90 ; 75 ; 75 )
```

## Result:

Returns "77BF30"

# MMColor_GetColorComponent

### Description

This function will return the value of a specific Component (eg. the Green component or the Luminance component) of the specified HexColor from the specified Color Space.

### Return Type

Number

### Format

MMColor_GetColorComponent ( **HexColor** ; **ColorSpace** ; **Component** )

### Required Parameters

**HexColor**
The Color to extract the Component from.

**ColorSpace**
The Color Spaces that contains the Component you want to retrieve.

**Component**
The Component to retrieve. (For Example, the "Green" Component.)

### Related Items

MMColor_GetColor, MMColor_SetColorComponent

### Examples

# Example 1

## Code:

```
MMColor_GetColorComponent( "12CC34" ; "RGB" ; "Green" )
```

## Result:

Returns 204

# Example 2

## Code:

```
MMColor_GetColorComponent( "12CC34" ; "HSL" ; "Luminance" )
```

## Result:

Returns 44

# MMColor_GetGrayscale

### Description

This function takes any Color you specify and returns the Grayscale equivalent of that color.

### Return Type

Text (HexColor)

### Format

MMColor_GetGrayscale ( **HexColor** )

### Required Parameters

**HexColor**
The Color you want to convert to Grayscale.

### Related Items

MMColor_IsWebSafe

### Example

## Code:

```
MMColor_GetGrayscale( "123456" )
```

## Result:

Returns "2E2E2E"

# MMColor_GetImageColor

### Description

This function will take an Image created with MMColor_CreateColorImage and return the associated color of that image. It will only work with Container Objects created directly with MMColor_CreateColorImage.

### Return Type

Text (HexColor)

### Format

MMColor_GetImageColor ( **Image** )

### Required Parameters

**Image**
A Container that holds the Image you want to retrieve the Color from. This only works for Images that were created with MMColor_CreateColorImage.

### Related Items

MMColor_CreateColorImage

### Example

## Code:

```
MMColor_GetImageColor( mydb::mycontainer )
```

## Result:

Returns "FFFF00" if the Image in the "mycontainer" field was created with MMColor_CreateColorImage and is a bright yellow color.

# MMColor_HexToDec

### Description

Returns the Decimal representation of the Hex number passed to the "Hex" parameter.

### Return Type

Number

### Format

MMColor_HexToDec ( **Hex** )

### Required Parameters

**Hex**
The Hexidecimal number to convert to Decimal.

### Related Items

MMColor_DecToHex

### Example

## Code:

```
MMColor_HexToDec( "7F" )
```

## Result:

Returns 127.

# MMColor_IsWebSafe

### Description

This function checks to see if the HexColor you specified will correctly be rendered on a webpage.

### Return Type

Number (1=True, 0=False)

### Format

MMColor_IsWebSafe ( **HexColor** )

### Required Parameters

**HexColor**
The Color to test.

### Related Items

MMColor_GetGrayscale

### Examples

# Example 1

## Code:

```
MMColor_IsWebSafe( "0099FF" )
```

## Result:

Returns 1.

# Example 2

## Code:

```
MMColor_IsWebSafe( "2300FF" )
```

## Result:

Returns 0.

# MMColor_Register

### Description

You can use this function to Register the plug-in from a script instead of through the Configuration Dialog. This is useful when the plug-in is being distributed to many computers, allowing you to intall and register the plug-in without having to physically visit each computer. This function also allows you to check if the plug-in is already registered or clear the current registration. The plug-in always requires you to accept the License Agreement to use the plug-in. This is usually done by presenting the License Agreement Dialog, but that can be suppressed by using the special option value "I Accept the License Agreement".

### Return Type

Text

### Format

MMColor_Register ( **FirstName** ; **LastName** ; **LicenseKey** ; **Option** )

### Required Parameters

**FirstName**
The Registration First Name you specified when you ordered. (See your Receipt.)

**LastName**
The Registration Last Name you specified when you ordered. (See your Receipt.)

**LicenseKey**
The License Key from your Receipt.

### Optional Parameters

**Option**
Specify "Dialog" to enter your Registration Information in a dialog.
Specify "Check" to see if the plug-in is already Registered.
Specify "Clear" to remove the Registration.
Specify "I Accept the License Agreement" to automatically accept the License Agreement dialog without showing it to the end user.
Notes: The "Dialog", "Check", and "Clear" options can also be specified as the first parameter. If you have a Developer License, do not use the "I Accept the License Agreement" value here. See your Developer Instructions file for more information.

### Examples

# Example 1

## Code:

```
MMColor_Register( "My First Name" ; "My Last Name" ; "My License Key" )
```

## Result:

Registers the plug-in with the provided registration information (obviously the above is not valid registration information; please see your Receipt).

# Example 2

## Code:

```
MMColor_Register( "Dialog" )
```

## Result:

Displays a dialog for you to enter your First Name, Last Name, and License Key as it appears on your Receipt.

# Example 3

## Code:

```
MMColor_Register( "Check" )
```

## Result:

Returns "Not Registered." or "Registered to <Name> for a <License>."

# Example 4

## Code:

```
MMColor_Register( "My Company Name" ; "My Company Name" ; "My Site License Key" ; "I Accep
```

## Result:

Registers the plug-in and uses the "I Accept the License Agreement" option to keep the License Agreement dialog from appearing.

# MMColor_SetColorComponent

### Description

This function updates the HexColor parameter you specify by adjusting the Value of the Component of the Color Space you specify. For example, you could set the Hue of the HSL Color Space of your HexColor.

### Return Type

Text (HexColor)

### Format

MMColor_SetColorComponent ( **HexColor** ; **ColorSpace** ; **Component** ; **Value** )

### Required Parameters

**HexColor**
The Color you need to modify.

**ColorSpace**
The Color Space that contains the Component you want to modify. (For Example, the "HSL" Color Space.)

**Component**
The Component to modify. (For Example, the "Hue" Component.)

**Value**
The new Value for the Component.

### Related Items

MMColor_GetColor, MMColor_GetColorComponent

### Examples

# Example 1

## Code:

```
MMColor_SetColorComponent( "D40000" ; "RGB" ; "Blue" ; 127 )
```

## Result:

Returns "D4007F".

# Example 2

## Code:

```
MMColor_SetColorComponent( "D40000" ; "HSL" ; "Hue" ; 90 )
```

## Result:

Returrns "69D400".

# MMColor_SetLayoutPartHeights

### Description

To properly display the color space sliders at specific pixel locations on layouts set to "List View", you need to let the plug-in know the heights of the Layout Parts of your layout. Once you have set these values, showing the color space sliders and have them appear to be attached to buttons in the list of records on your layout is as easy as specifying the pixel locations of the buttons on your layout in layout mode. See the MMColor_ShowColorSpaceSlider and MMColor_ShowSlider functions for more information about showing menus at specific locations.

### Return Type

Text

### Format

MMColor_SetLayoutPartHeights ( **Header** ; **Body** ; **Footer** )

### Required Parameters

**Header**
The Height of the Header Part.

**Body**
The Height of the Body Part.

**Footer**
The Height of the Footer Part.

# MMColor_ShowColorPicker

### Description

This function will show the system Color Picker allowing the user to select a color from it. It returns the HexColor representation of the chosen color. To start the Color Picker with a specific color, specify the StartHexColor parameter. If the user cancels the Color Picker, the StartHexColor is returned if it was specified, otherwise it returns nothing.

### Return Type

Text (HexColor)

### Format

MMColor_ShowColorPicker ( **StartHexColor** )

### Optional Parameters

**StartHexColor**
The initial Color for the Color Picker dialog.

### Example

## Code:

```
MMColor_ShowColorPicker( "00FFFF" )
```

## Result:

Shows the system Color Picker dialog starting with a Cyan color.

# MMColor_ShowColorSpaceSlider

### Description

This function will bring up a dialog with 3 to 4 Sliders (depending on the Color Space) allowing the user to manipulate the sliders or type in literal values to get the exact color they want. The function then returns that color as a HexColor. The optional AttachToObject parameter will allow you to attach the Slider dialog to an object on your Layout in much the same way a Drop Down List can be attached to a field.

### Return Type

Text (HexColor)

### Format

MMColor_ShowColorSpaceSlider ( **HexColor** ; **ColorSpace** ; **AttachToObject** )

### Required Parameters

**HexColor**
The Color you are adjusting.

**ColorSpace**
The Color Space to adjust with the Sliders.

### Optional Parameters

**AttachToObject**
If you specify "True", the slider will attach itself to the active field. If you specify an object name, it will attach itself to that object in FM 8.5 and above."

### Related Items

MMColor_ShowSlider

### Examples

# Example 1

## Code:

```
MMColor_ShowColorSpaceSlider( "FF00FF" ; "RGB" )
```

## Result:

Brings up the Color Space Slider dialog at the current Mouse Cursor location and allows you to adjust the sliders. It then returns whatever color was chosen.

# Example 2

## Code:

```
MMColor_ShowColorSpaceSlider( "00FF80" ; "HSL" ; "ChooseColorButton" )
```

## Result:

Brings up the Color Space Slider dialog aligned to the bottom and left of the Layout Object named "ChooseColorButton". It returns whatever color was chosen.

# MMColor_ShowSlider

### Description

This function will bring up a dialog with a Sliders and Edit field allowing the user to select a value for a Color Space Component. By specify the Color Space and the Component parameters, the Slider will automatically adjust to only allow Values in the correct range for that Component. The InitialValue tells the Slider where to begin when the dialog opens. The optional AttachToObject parameter will allow you to attach the Slider dialog to an object on your Layout in much the same way a Drop Down List can be attached to a field. If you specify the optional HexColor parameter, a Color chip will also appear allowing the user to see in realtime how their adjustments are changing the color.

### Return Type

Number

### Format

MMColor_ShowSlider ( **ColorSpace** ; **Component** ; **InitialValue** ; **AttachToObject** ; **HexColor** )

### Required Parameters

**ColorSpace**
The Color Space to that contains the Component you want to adjust.

**Component**
The Component to adjust with the Slider.

**InitialValue**
The Initial Value of the Component you are adjusting. (This value is also returned if the Slider is canceled with the Esc key.)

### Optional Parameters

**AttachToObject**
If you specify "True", the slider will attach itself to the active field. If you specify an object name, it will attach itself to that object in FM 8.5 and above."

**HexColor**
The Color you are adjusting. (If you specify this parameter, the Slider will include an updating color block.)

### Related Items

MMColor_ShowColorSpaceSlider

### Examples

# Example 1

## Code:

```
MMColor_ShowSlider( "RGB" ; "Blue" ; 127 )
```

## Result:

Brings up the Slider dialog at the current Mouse Cursor location and allows you to adjust the Blue Component. It then returns whatever Blue value was chosen.

# Example 2

## Code:

```
MMColor_ShowSlider( "HSL" ; "Hue" ; 323 ; "AdjustHueButton" ; "D4007F" )
```

## Result:

Brings up the Slider dialog aligned to the bottom and left of the Layout Object named "AdjustHueButton". The dialog will have a Color chip on it allowing the user to see the change to the Hue value in realtime. It then returns whatever new Hue was chosen.

# MMColor_Version

### Description

This function returns the current version of MMColor. This function is useful for testing whether or not the plug-in is installed and enabled. If you call this function and a question mark ("?") is returned, then the plug-in is either not installed or not enabled.

### Return Type

Text

### Format

MMColor_Version

### Related Items

MMColor_VersionAutoUpdate

### Example

## Code:

```
MMColor_Version
```

## Result:

Returns the MMColor version like "MMColor v.1.6.1".

# MMColor_VersionAutoUpdate

### Description

This function returns an Auto Update friendly Version number of MMColor. The format of this version number is always exactly 8 digits long. The first two digits represent the major version of the plug-in (zero-filled). The third and fourth digits represent the minor version of the plug-in (zero-filled). The fifth and sixth digits represent the update portion of the version (zero-filled). The final two digits represent a special build number or a beta version number and will usually be zeros.

As an example, for MMColor 1.6.1, the major version is 1, the minor version is 6, the update number is 1, and there is no special build or beta version defined. So, the resulting Auto Update friendly version number would be 01060100.

### Return Type

Number

### Format

MMColor_VersionAutoUpdate

### Related Items

MMColor_Version

### Example

## Code:

```
MMColor_VersionAutoUpdate
```

## Result:

Returns 01060100 for MMColor version 1.6.1.